

# Weekly Report

08/21/2015-09/27/2015

## Research

### 1 覆盖基站数据

这周把六个城市（温州、宁波、湖州、台州、舟山和嘉兴）的范围比较大的绿地和商业区的范围，并且计算出覆盖它们的基站。数据已经处理完成并存储在电脑上，其中温州的数据是暑假做的，当时已发给师兄。

### 2 MatrixWave: Visual Comparison of Event Sequence Data

周四报告了这篇关于比较两个时间序列事件的可视化文章。事件序列是由一系列包含时间属性的事件组成，每一条事件序列包含了按时间排序的多个事件。网站的点击流数据也是一种事件序列数据。当用户访问网站时，他们在多个页面之间跳转的过程，可以作为分析人员分析这个网站浏览情况的一种数据。一直以来，桑基图（Sankey Diagram）是可视化这类数据的常用可视化方法。但是随着变量增多，跳转关系变复杂，桑基图就开始出现视觉遮挡（图2）。为了解决这个问题，作者提出了MatrixWave的可视化方法，用矩阵来代替节点之间的关系，使之能够可视化更大更密集的事件序列数据。

#### 2.1 MatrixWave结构

网站分析者通常比较关心用户在页面间的跳转关系，所以这篇文章忽略了事件的时间属性，只保留了时间上的先后逻辑关系。图2.1-a 是提炼出来的网站点击流数据，第一行的用户从网页A跳转到网页D再跳转到网页B，最后浏览完了网页B后关闭了整个网页。整个流程可以由桑基图2.1-b展示。矩形结点表示特定网页，两个结点之间的连接表示网页之间的跳转。结点的大小表示该网页的流量，连接的

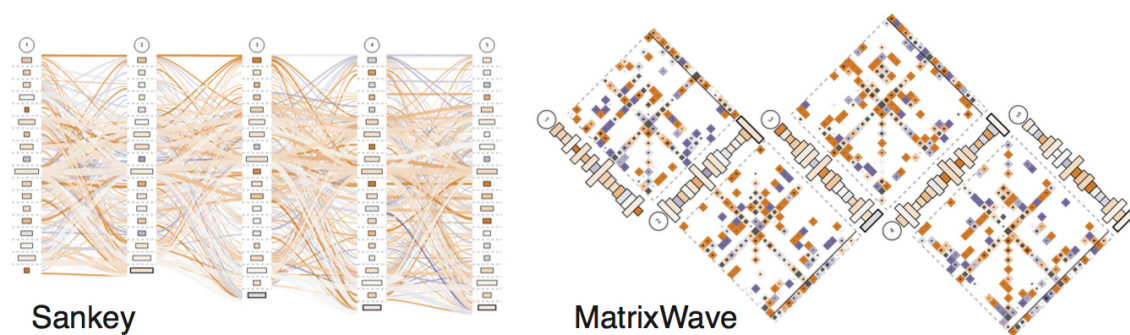


Figure 1: 时间序列数据和桑基图

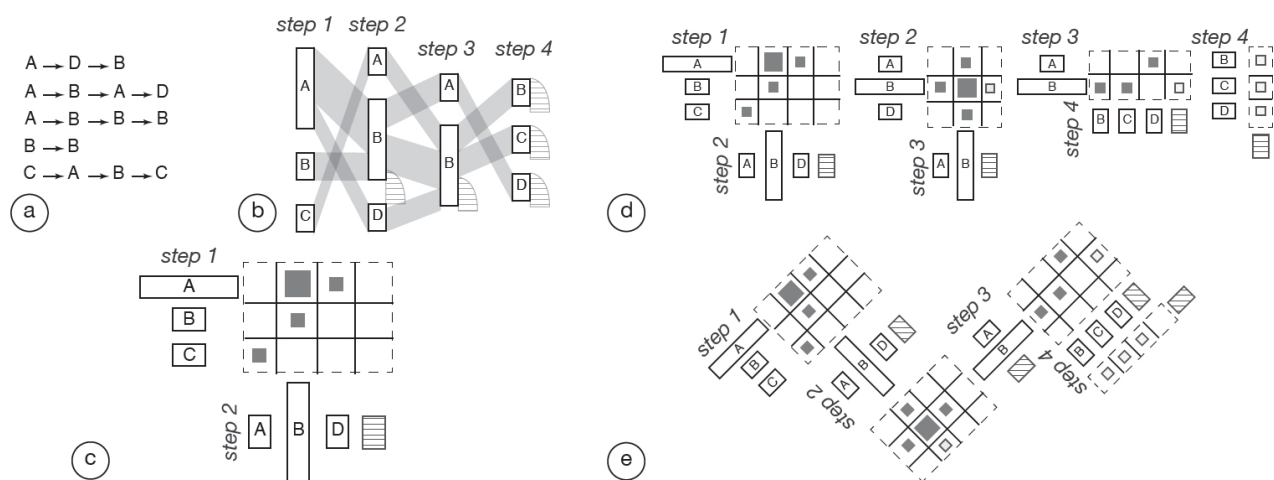


Figure 2: matrixwave可视化结构

较密集的时候，桑基图会出现视觉遮挡。为了克服这个问题，MatrixWave把结点间转移的关系用矩阵来表示，矩阵中的每个单元表示了结点间的转移流量。图2.1-c展示了Step1到Step2的跳转。同样，结点的大小表示网页的流量，矩阵中方形的大小表示网页间跳转的流量。为了展示用户关闭网页的动作（对于分析用户行为有所帮助），增加了一个离开结点，由阴影表示。由此，相邻的步骤之间就可以用矩阵代替，构成了图2.1-d。为了展示用户浏览整个网站的路径，我们用之字形的走法把矩阵按照先后顺序连接起来，我们就可以在图2.1-e中观察用户的浏览过程。

## 2.2 用于比较的可视化编码

常用的可视化比较方法有四种：并列（juxtaposition）、重叠（superposition）、显示编码（explicit encoding）和动画（animation）。如图2.2所示，X和Y是两个数据，（a）（b）（c）依次是并列、重叠和显示编码方法。并列方法要求用户在两个数据之间不断比较，这里就要求用户记住另一个的数据位置，对用户来说不够方便。重叠是把两个数据放在一起，让用户比较，相比于并列方法更加简单，但是在元素比较多的情况下，图详解就会比较混乱。显示编码直接显示了两个数据的差别，更为直观。动画方法需要把动画一遍一遍不断播放，同样不是很方便。文章中，作者采取了显式编码这种视觉比较方法。在这里，我们需要编码两个信

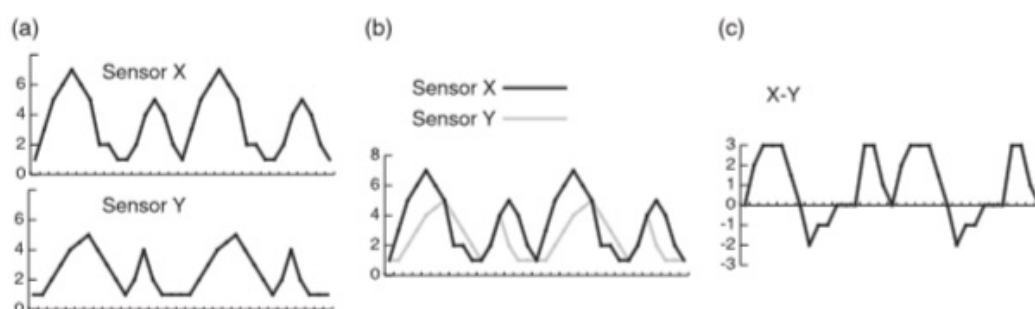


Figure 3: matrixwave可视化结构

息，一是两个数据集的绝对流量，二是两个数据集比较的流量，这里称之为相对流量。如图2.2，对于结点，我们用大小表示两个数据集的绝对流量（同一个网页在两个数据集中流量的平均值），同时，用颜色编码相对流量，当网页在数据集1中的流量比较大时用紫色表示，当在数据集2中流量比较大时用橙色表示。类似的，对于矩阵单元，其内部方形的大小表示两个数据集中相应两个节点之间的平均转移流量，颜色采用了与结点一样的颜色方案来表示转移流量在两个数据中的差异。



Figure 4: 比较可视化编码

## 2.3 用户调研

为了评估MatrixWave的有效性，作者还开展了一个用户调研来比较MatrixWave和桑基图这两种表示形式。结果显示，相比于辛基图，使用MatrixWave可以获得更高的准确度（95% VS 79%）和更少的完成时间（34.0s VS 38.2s）。MatrixWave相比于传统的桑基图能够有效地展示大且密集的事件序列数据集，桑基图则可以用来展示数据量比较小的数据，因为桑基图更容易被用户接受。然而由于MatrixWave的布局是之字形的，把矩阵旋转过45度之后不利于用户的观察分析。

## 3 Latent Dirichlet Allocation

这周看了一些LDA的东西，发现原文看不懂之后，又在网上找到一篇LDA数学八卦。这篇50多页的文档阐述了LDA模型所用到的大部分知识Dirichlet分布、多项分布、Gibbs抽样。LDA的原始文献，推测模型的参数用的是变分推断和EM算法，在之后的应用中，开始使用Gibbs采样，直接采样参数。这里介绍一下模型的思路，之后看懂了知识至今的联系之后再做一个详细的报告。

LDA模型中，每一篇文章都是一个词袋，就是说不考虑词语间的先后顺序，把每篇文章看做一个词频分量。假设所有的单词都构成词汇表 $V = \{v_1, v_2, \dots, v_n\}$ ，所有主题都构成一个主题表 $T = \{t_1, t_2, \dots, t_K\}$ ，每一个主题都对应一个多项式概率分布的词汇表，比如关于计算机的主题那么计算机词汇的概率就会高一点，诗歌散文的词汇就会低一点。所以每次生成文档时，都要选择一个主题分布（我们认为一篇文章应该有多个主题，比如一篇计算机文章80%关于计算机，20%关于数学），根据主题分布得到一个词汇分布，根据词汇分布得到一个词汇。

对于语料库中的每篇文档，LDA定义了如下生成过程：

对于每一篇文章，首先选择一个主题分布。重复以下过程，得到一篇文章

- 1.由主题分布随机得到一个主题（一个主题对应了一个单词分布）
- 2.由单词分布得到一个单词

语料库中的每一篇文档与  $K$  个主题的一个多项分布相对应，将该多项分布记为 $\theta$ 。每个主题又与词汇表中的  $V$  个单词的一个多项分布相对应，将这个多项分布记为 $\phi$ 。 $\theta$  和 $\phi$ 都有一个先验假设 $\alpha$ 和 $\beta$ ，这是贝叶斯后验概率所要求的。这里也是我所不明白的，为什么要一个先验假设，我猜测是为了Gibbs采样中边缘概率公式的推导。

整个模型最后就是要学习两个参数，一个是文档-主题分布 $\theta$ ，另外是主题-单词分布 $\phi$ 。推断方法主要有LDA模型作者提出的变分-EM算法，还有现在常用的Gibbs抽样法。

## Plan for next week

- 把以前Mobility Pattern的程序跑起来，之前为了做查找覆盖基站的任务，好像改了一些参数。
- 继续学习LDA算法。
- 操作系统和体系结构等课程本科没有学过，需要花点时间大概了解一下。